

特開平9-37249

(43)公開日 平成9年(1997)2月7日

(51) Int.CI.
H04N 7/24
H03M 13/00

識別記号 記入欄
H04N 7/13
H03M 13/00

F I
H04N 7/13
H03M 13/00

技術表示箇所 A

審査請求 未請求 請求項の数 14 O L (全19頁)

(21)出願番号 特願平7-184445
(22)出願日 平成7年(1995)7月20日

(71)出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地
(72)発明者 溝添 博樹
神奈川県横浜市戸塚区吉田町292番地
株式会社日立製作所映像メディア研究所内
(72)発明者 奥 万寿男
神奈川県横浜市戸塚区吉田町292番地
株式会社日立製作所映像メディア研究所内
(72)発明者 三代 隆之
東京都小平市上水本町5丁目20番1号
株式会社日立製作所半導体事業部内
(74)代理人 弁理士 武 頭次郎

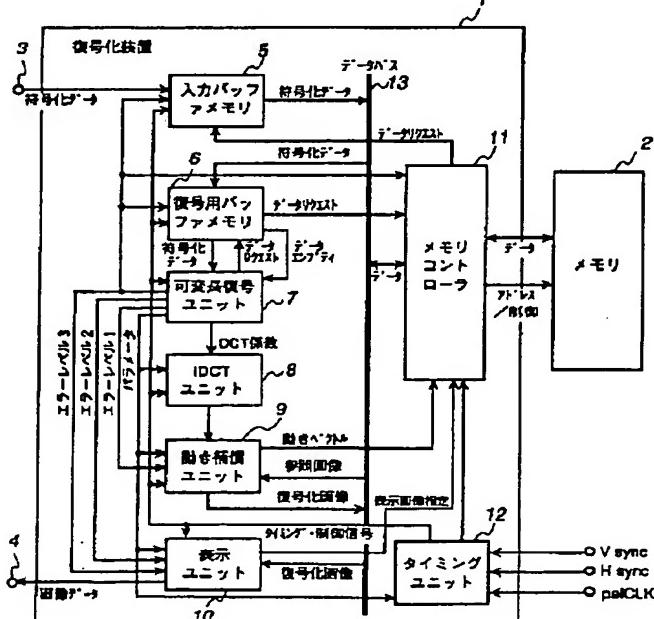
(54)【発明の名称】符号化映像信号の復号化処理方法及びそれを用いた復号化装置

(57)【要約】

【目的】発生するエラーの重大性に対応したレイヤへの復号化処理に速やかに復帰させ、エラーに伴なう画像の乱れ時間を必要最低限に抑えるようにする。

【構成】符号化データは、入力バッファメモリ5を介して外部メモリ2に記憶され、そこから読み出されて、復号用バッファメモリ6を開始、可変長復号ユニット7、IDCTユニット8、動き補償ユニット9で復号化処理され、復号化画像として外部メモリ2に記憶されるとともに、そこから読み出されて表示ユニット10で処理され、表示に供される。可変長復号ユニット7では、復号処理において、エラーが発生すると、その重大性の程度に応じて評価し、この評価結果に応じたエラーレベル1、2、3の信号を出力する。このエラーレベル1、2、3に応じて、符号化データでの復号処理動作を画像の始めから開始させるか、画像中で開始させるなど、その処理開始段階を異ならせる。

【図1】



【特許請求の範囲】

【請求項 1】 符号化された映像信号の復号化処理方法において、

復号化処理中に発生するエラーを検出し、

検出される該エラーの重要性の程度を複数のレベルに評価し、

エラーから正常な復号化処理への復帰の段階をその評価レベルに応じて異ならせるこことを特徴とする符号化映像信号の復号化処理方法。

【請求項 2】 請求項 1において、

前記符号化映像信号の符号列中に挿入されたエラーコードを検知することにより、前記エラーの発生を検出することを特徴とする符号化映像信号の復号化処理方法。

【請求項 3】 請求項 1において、

前記符号化映像信号の符号列の中から正規の符号として予め定義された以外の符号を検知することにより、前記エラーの発生を検出することを特徴とする符号化映像信号の復号化処理方法。

【請求項 4】 請求項 1において、

前記符号化映像信号の特定の種類の符号が夫々異なるテーブルによって復号化され、
これらテーブルの少なくとも 1 つで、そこに定義されていない符号が入力されたことを検知することにより、前記エラーの発生を検出することを特徴とする符号化映像信号の復号化処理方法。

【請求項 5】 請求項 1において、

前記符号化映像信号を復号化して得られたパラメータの値が予め定義された範囲以外であることを検知することにより、前記エラーの発生を検出することを特徴とする符号化映像信号の復号化処理方法。

【請求項 6】 請求項 1において、

前記検出エラーは、その検出によって第 1 番目のエラーレベルと評価し、

第 n 番目（但し、n = 1, 2, 3, ……）のエラーレベルと評価されたエラーは、このエラーから前記正常な復号化処理へ復帰する前に復号タイミングに同期したパルス信号を検知したとき、第 n + 1 番目のエラーレベルと評価することを特徴とする符号化映像信号の復号化処理方法。

【請求項 7】 請求項 1において、

前記エラーの検出毎に計数値が 1 ずつ増加するエラーの発生回数の計数を行ない、その計数値に応じたレベルとしてそのとき検出した該エラーを評価することを特徴とする符号化映像信号の復号化処理方法。

【請求項 8】 符号化された映像信号の復号化装置において、

復号化中の該符号化映像信号のエラーを検出するエラー検出手段と、

該検出エラーの重要度の程度を複数のレベルに評価するエラー評価手段と、

該エラーから正常な復号化処理への復帰の段階をその評価レベルに応じて異ならせる復号化処理制御手段とを具備したことと特徴とする符号化映像信号の復号化装置。

【請求項 9】 請求項 8において、

前記エラー検出手段は、前記符号化映像信号の符号列中に挿入されたエラーコードを検知することによってエラー発生を検出することを特徴とする符号化映像信号の復号化装置。

【請求項 10】 請求項 8において、

10 前記エラー検出手段は、前記符号化映像信号の符号列の中から正規の符号として予め定義された以外の符号を検知することによってエラー発生を検出することを特徴とする符号化映像信号の復号化装置。

【請求項 11】 請求項 8において、

前記符号化映像信号の特定の種類の符号を復号化するためのテーブルを夫々の符号に対応して複数個備え、
復号化しようとする該符号に対応して該テーブルの中から適切な 1 つが前記復号化処理制御手段によって選択され、

20 前記エラー検出手段は、該テーブルの少なくとも 1 つでそこに定義されていない符号の入力があったとき、エラーが発生したと判断することを特徴とする符号化映像信号の復号化装置。

【請求項 12】 請求項 8において、

前記エラー検出手段は、前記符号化映像信号を復号化して得られたパラメータの値が予め定義された範囲以外であることを検知することによってエラー発生を検出することを特徴とする符号化映像信号の復号化装置。

【請求項 13】 請求項 8において、

30 前記エラー評価手段は、
前記エラーを、それが前記エラー検出手段で検出されることにより、第 1 番目のエラーレベルと評価し、
第 n 番目（但し、n = 1, 2, 3, ……）のエラーレベルと評価したエラーを、このエラーから前記正常な復号化処理へ復帰する前に復号タイミングに同期したパルス信号を検知したとき、第 n + 1 番目のエラーレベルと評価することを特徴とする符号化映像信号の復号化装置。

【請求項 14】 請求項 8において、

前記エラー評価手段は、

40 前記エラー検出手段でエラーが検出される毎にこれを計数して計数値を 1 ずつ増加させるカウンタと、
前記エラー検出手段で検出されるエラーを該計数値に応じたレベルに評価する手段とを備えたことを特徴とする符号化映像信号の復号化装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、データ量削減のためなどで符号化された映像信号の復号化処理方法及びそれを用いた復号化装置に関する。

【従来の技術】ディジタル化された映像信号は、特に、それが動画像の映像信号である場合、膨大な情報量を有している。そのため、かかる映像信号を磁気ディスクなどの記録媒体に長時間にわたってそのまま記録しようとすると、非常に大きい記憶容量が必要になり、コストがかさむことになる。また、有線や無線によってリアルタイムで伝送しようとすると、映像データのビットレートが高いことから、非常に広帯域な伝送路が必要であり、その実現は容易でない。そこで、従来、信号処理により映像データを効率良く符号化してデータ量を削減するための符号化方式がいくつか提案されている。

【0003】のような符号化方式として、MPEG1（「OPTRONICS」；1992-No.5 pp.86～98参照）及びMPEG2（「テレビジョン学会誌」；Vol.48 No.1 pp.44～49参照）がある（以下、単にMPEGと表現した場合には、両者を指すものとする）。MPEGでは、映像データのデータ量を削減するために、主に、以下の複数の方法が適宜組み合わせて用いられる。

【0004】第1の方法は、符号化する画像とその前後の画像（以下、これを参照画像という）との間で差分を取りて振幅の小さい情報信号に変換することにより、符号化に要するビット数を少なくする方法である。

【0005】MPEGでは、符号化を行なう画像の単位をピクチャという。MPEG1の場合、1ピクチャが原画像の1フレームで構成されている。また、MPEG2の場合には、1ピクチャが原画像の1フレーム単位または1フィールド単位で構成されており、どちらの単位を用いるかは符号化時に選択可能となっている。

【0006】画像の参照方法により、I、P、Bピクチャの3種類のピクチャが存在する。これを図13に示す。

【0007】同図において、矢印の始点となる画像が参照画像、終点となる画像が復号中のピクチャを表わしている。Iピクチャは画像の参照を行なわない。これは、復号化のために必要な情報が全てそのピクチャ内に符号化されているからである。Pピクチャは直前に復号化したIピクチャまたはPピクチャを参照画像とする。Bピクチャはその直前と直後に存在するIまたはPピクチャを参照画像とする。

【0008】1ピクチャは、マクロブロックという一定の大きさを持つ区画で区切られている。さらに、マクロブロックは8画素×8画素の単位画像（以下、ブロックという）から構成されている。例えば、カラー映像信号がY:Cb:Cr=4:1:1の形式である場合、図14(a)に示すように、輝度信号がY0～Y3の4つのブロック、色差信号Cb、Crの各1つずつのブロック、計6つのブロックによってマクロブロックが構成されている。

【0009】これらマクロブロックには、図14(b)に示すように、画像の左上隅から順にマクロブロックア

ドレスと呼ばれる通し番号が付けられている。ここで、例えば、1ピクチャが 720×480 画素のフレーム画像から構成されている場合には、そこに、 $(720 \times 480 \text{ 画素}) \div (8 \times 8 \times 4) = 1350$ 個のマクロブロックが存在する。従って、順次のマクロブロックに0～1349のマクロブロックアドレスが設定される。

【0010】マクロブロックは、参照画像との差分を取る最小単位である。画像の参照の方法はマクロブロック単位で変えられるようになっている。そのため、それを示す情報及び参照画像中で実際に参照に用いた部分を示す情報（これを動きベクトル情報という）を各マクロブロックの符号化データ中にエンコードする。

【0011】データ量削減のための第2の方法は、映像データにDCT処理（離散コサイン変換処理）を施すことにより、空間周波数領域に変換する方法である。映像信号の場合には、一般に、空間周波数の低域成分にエネルギーが集中している割合が高く、高域成分は振幅が小さい。従って、このことから、符号化に割り当てるビット数を低域成分ほど多くし、高域成分ほど少なくすることにより、ビット数の割当てを最適化することができ、この結果、全体のビット数を削減することが可能である。MPEG1、2では、上記のブロック単位でDCTを行ない、データの削減を図っている。

【0012】データ量削減のための第3の方法は、同じ値のデータが複数個連続している場合、かかる同じ値のデータを繰り返し送る代わりに、その値とそのデータの個数とを表わすデータを送ることにより、データ量を削減する方法である。MPEGでは、DCT係数の符号化に際してこの方法を用いている。

【0013】データ量削減のための第4の方法は、夫々の値の出現確率に応じて異なる長さの符号を割り当てる可変長符号化である。出現頻度の高い値ほど短い符号を割り当てるこにより、全体のビット数を削減することが可能である。MPEGでは、DCT係数のほかに、マクロブロックアドレスや動きベクトルなどのパラメータの符号化に可変長符号化を用いている。これらは夫々、可変長符号と復号すべき値の対応表が予め規格の中に定められている。

【0014】次に、符号の階層構造について説明する。

【0015】MPEGの符号化データはレイヤと呼ばれる階層構造を有しており、図15に示すように、シーケンスレイヤからブロックレイヤまでの6つの階層がある。

【0016】最上位層のシーケンスレイヤはシーケンスヘッダから始まり、1つ以上のグループ・オブ・ピクチャレイヤを含み、シーケンスエンドコードで終了する符号の単位である。シーケンスヘッダには、ピクチャサイズやフレームレートなど一連のシーケンスレイヤに共通なパラメータが符号化されている。なお、チャネル切替えなどのように、シーケンスレイヤの途中から復号を開

始する場合に備えて、シーケンスレイヤのヘッダは符号化側でシーケンス中に適宜繰り返し挿入可能となっている。

【0017】上記グループ・オブ・ピクチャレイヤはグループ・オブ・ピクチャヘッダから始まり、複数のピクチャレイヤを含んでいる。

【0018】このピクチャレイヤは1つのピクチャを含むものであり、ピクチャヘッダから始まって複数のストライスレイヤを含んでいる。先に説明したように、1つのピクチャは1枚のフレーム画像または1つのフィールド画像に対応する。ピクチャヘッダには、そのピクチャがI, P, Bピクチャのいずれであるかを区別するパラメータなどが符号化されている。

【0019】上記ストライスレイヤはストライスヘッダから始まり、複数のマクロブロック毎のマクロブロックレイヤを含んでいる。このストライスヘッダには、ストライスの画面上の垂直位置を表わす情報が符号化されており、また、ストライスヘッダ直後のマクロブロックレイヤには、このストライスの画面上の水平位置を表わす情報が符号化されている。そして、これら2つの情報を合わせると、マクロブロックアドレスの絶対アドレスを求めることができる。

【0020】上記マクロブロックレイヤはマクロブロックを構成するブロックを含むものであって、マクロブロックヘッダと、図14(a)に示したマクロブロックを構成する6つのブロック毎のレイヤ、即ち、ブロックレイヤとを含んでいる。マクロブロックヘッダには、先の動きベクトル情報やマクロブロックアドレスを示す情報などが符号化されている。

【0021】最下位層の上記ブロックレイヤは、ブロック中の各画素に対するDCT係数を符号化して含んでいる。

【0022】以上の各レイヤのうち、シーケンスレイヤからストライスレイヤまでのストライスレイヤ以上の層のヘッダは、スタートコードと呼ばれる符号から開始する。スタートコードは3バイト(=24ビット)の長さを有する符号であって、スタートコード以外にそれと同じパターンが符号化データ中に現れることが禁止されている。また、符号化データを先頭から1バイト単位で区切ったとき、スタートコードの先頭は丁度その区切りから開始するように配置されている(バイトアラインメント)。このことから、万一エラーなどによって符号化データの連続性を見失ったような場合でも、スタートコードの発見は容易であり、それを手がかりにして、少なくともストライスレイヤ以上の層のヘッダで正常な復号処理へ復帰させることができる。

【0023】ところで、MPEGにおいては、符号化データ中に誤りが含まれていることを示すシーケンスエラーコードが用意されている。これは、例えば、符号化データの伝送途中で発生した誤りを訂正しきれなかった場

合、伝送媒体によって挿入されるものである。しかし、MPEGの規格では、シーケンスエラーコード以外にエラーの発生を示す手段については規定されていない。また、様々な種類が考えられるエラーに対して、それらを区別したり、エラーの重大性の度合を評価したりすることについても規定されていない。さらに、実際にエラーが発生した場合の対処の仕方や復帰方法についても触れていない。

【0024】

10 【発明が解決しようとする課題】伝送途中での誤りの発生や符号化の際のミスなどの何らかの原因により、符号化映像信号の復号中に、エラーが発生することがあり得る。かかるエラーを検出した場合、先に説明したスタートコードを検索し、それを手がかりにして正常な復号処理への復帰を図るようにすることは可能である。しかし、発生したエラーの重大さ(エラーレベル)を評価せず、常に同一の過程を経て正常な復号処理へ復帰を図るようにしていると、以下のような問題が生ずる。

【0025】即ち、例えば、エラー発生後、常にストライスレイヤ以上の層のスタートコードの検出で復帰を図るようなシステムにおいて、重大なエラーが発生した場合を考える。ここでいう重大なエラーとは、例えば、チャネル切替えで全く別の符号化データに瞬間に切り替わった場合のように、符号化データの連続性が失われている場合である。

【0026】もしも、ピクチャサイズのように、チャネル切替えの前後で復号に不可欠な重要なパラメータが異なっていると、実際上それ以上復号を継続するのは困難である。それにもかかわらず、このシステムは、エラーレベルを評価する手段を持たないので、スタートコードを有する最下位層であるストライスレイヤのストライスヘッダのスタートコードを検出する度に正常な復号処理へ復帰してしまう。そして、その後復号処理に破綻を来たし、再びエラー処理に入るという繰返しが行なわれる。そして、その結果、著しい画像の劣化を引き起こすことになる。

【0027】また、例えば、エラー発生後、常にシーケンスヘッダのスタートコードの検出で復帰を図るようなシステムを考える。このようなシステムでは、前述のような重大なエラーが発生した場合でも、次のシーケンスヘッダを検出するまで復号処理に復帰しないので、それまでの間、例えば、直前の参照画像を表示し続けるようにしておけば、画像の乱れは最小限に留めることができる。

【0028】ところが、このようなシステムにおいては、比較的軽微なエラーが発生した場合に問題が生じる。ここでいう軽微なエラーとは、例えば、DCT係数の数ビットが誤っている場合である。

【0029】このような場合、エラーの影響を受けるのはせいぜいその誤りを含むストライスのみであるので、次

7

8

のスライスですぐに復帰すれば、以後は正常に復号処理を続けていくことが可能である。それにもかかわらず、このシステムは、エラーレベルを評価する手段を持たないので、次のシーケンスヘッダでのスタートコードを検出するまでは復帰しない。そのため、次のスライスで復帰する場合に比べて正常な復号処理への復帰が遅れ、正常な画像を表示することが長時間できなくなってしまう。

【0030】以上のように、エラーレベルを評価せず、エラー検出から正常な復号処理への復帰を常に同じ方法で行なうようにしていると、エラーレベルに応じた適切なレイヤへ必要最低限の時間で復帰することができず、その結果、正常でない画像を長時間表示してしまう可能性がある。

【0031】本発明の目的は、エラーレベルに応じた適切なレイヤで速やかに復号動作を復帰させ、エラーによる正常でない画像を表示する時間を必要最低限の時間に抑えることを可能とした符号化映像信号の復号化処理方法及びそれを用いた復号化装置を提供することにある。

【0032】

【課題を解決するための手段】上記目的を達成するために、本発明は、復号化時にエラーを検出すると、この検出したエラーの程度を複数のレベルに評価し、その評価に基づいてエラーからの復帰過程を切り替えるようとする。

【0033】

【作用】符号化映像信号の復号中に符号のエラーがあった場合、このエラーが検出されると、この検出エラーがその重要度を表わすレベルのいずれに該当するか評価される。この評価の結果に応じて、正常な復号処理への復帰の段階を異ならせる。即ち、検出エラーの評価レベルが低いときには、符号化映像信号の下位の階層から復帰するようにし、検出エラーの評価レベルが高いときは、上位の階層から復帰するようにする。

【0034】このようにして、検出エラーの評価レベルに応じて正常な復号処理への復帰の段階を異ならせることにより、エラーの評価レベルに応じた適切なレイヤで速やかに復帰を図ることができる。

【0035】

【実施例】以下に本発明の実施例を図面を用いて説明する。なお、以下に説明する実施例では、入力信号として先に説明したMPEGに準ずる映像信号の符号化データを想定する。

【0036】図1は本発明による符号化映像信号の復号化処理方法及びそれを用いた復号化装置の一実施例を示すブロック図であって、1は復号化装置、2は外部メモリ、3は入力端子、4は出力端子、5は入力バッファメモリ、6は復号用バッファメモリ、7は可変長復号ユニット、8はIDCT(逆離散コサイン変換)ユニット、9は動き補償ユニット、10は表示ユニット、11はメ

モリコントローラ、12はタイミングユニット、13はデータバスである。

【0037】同図において、MPEGに準ずる映像信号の符号化データが入力端子3から入力され、入力バッファメモリ5に一時格納される。メモリコントローラ11からデータリクエスト信号があると、入力バッファメモリ5から符号化データが読み出され、データバス13を介してメモリコントローラ11に送られる。メモリコントローラ11はこの符号化データをアドレス及び制御信号とともに外部メモリ2に送り、この符号化データを外部メモリ2の所定の領域に記憶させる。

【0038】また、メモリコントローラ11は、復号用バッファメモリ6からデータリクエスト信号があると、アドレス及び制御信号により、外部メモリ2から符号化データを読み出し、復号用バッファメモリ6に転送する。この復号用バッファメモリ6に一時格納された符号化データは可変長復号ユニット7、IDCTユニット8及び動き補償ユニット9によって復号される。

【0039】この復号化されたフレームまたはフィールド単位の画像(復号化画像)は、データバス13を介してメモリコントローラ11に供給され、これにより外部メモリ2の所定の領域に記憶される。外部メモリ2に記憶された復号化画像は、一方では、表示画像として読み出され、データバス13を介して表示ユニット10に供給され、所定の表示処理がなされて出力端子4から出力される。また、他方では、PピクチャやBピクチャのように復号化処理に参照画像が必要な場合、メモリコントローラ11は、上記のように、外部メモリ2から復号化処理すべきPまたはBピクチャの符号化データを読み出して復号用バッファメモリ6に転送するとともに、外部メモリ2から復号化画像を読み出し、参照画像としてデータバス13を介して動き補償ユニット9に転送する。

【0040】また、MPEGの規格に従う符号化映像信号の場合、先に説明したスタートコードのパターンの特有性から、符号化データ中にゼロが連続して5バイト分以上存在した場合には、そのうち少なくとも1バイト分はダミーで挿入されたゼロデータであることが分かっている。ゼロが連続して5バイト見つかった場合には、このことを利用して、入力バッファメモリ5では、機械的にそのうちの1バイト分を除去するようにして符号化データの出力を行ない、後段の復号化処理部における処理負担の低減化を図っている。

【0041】復号用バッファメモリ6は、その内部に空きが生じると、メモリコントローラ11にデータリクエスト信号を出し、このメモリコントローラ11の制御によって外部メモリ2から転送される符号化データを受け取って蓄積する。そして、可変長復号ユニット7からデータリクエスト信号を受けると、それに応じて内部に蓄積したデータを出力する。但し、外部メモリ2内に符号化データが残っていない場合には、データエンブティ

信号を出力して復号用バッファメモリ 6 が空であること
を可変長復号ユニット 7 に知らせる。

【0042】可変長復号ユニット 7 は、後に詳細に説明するが、復号用バッファメモリ 6 から符号化データを受けてそれを解析し、符号化データ中の上記のようなパラメータ及び D C T 係数を復号化する。このパラメータは、復号化処理に必要な情報として復号化装置 1 内の各復号化処理ブロック 8 ~ 10, 12 へ送られる。また、D C T 係数は I D C T ユニット 8 へ送られる。

【0043】I D C T ユニット 8 は、可変長復号ユニット 7 から受けた D C T 係数データを逆 D C T 処理し、それによって得られる I D C T 係数を動き補償ユニット 9 に送る。

【0044】動き補償ユニット 9 は、まず、可変長復号ユニット 7 で復号化したパラメータから動きベクトルを再生し、それをメモリコントローラ 11 に与えて外部メモリ 2 に記憶されている復号化画像のうちから所定の復号化画像を参照画像として読み出させる。そして、この参照画像と I D C T ユニット 8 の出力とを加算し、復号化画像を再生する。再生された復号化画像は、表示用画像や次の画像の復号化処理のための参照画像として、メモリコントローラ 11 を介して外部メモリ 2 内に記憶される。

【0045】表示ユニット 10 は、外部メモリ 2 からメモリコントローラ 11 を介して表示タイミングに合わせて復号化画像を読み出し、最終的な画像データに変換した後、出力端子 4 から出力する。

【0046】タイミングユニット 12 には、表示用画素クロック (pelCLK) と、これに同期した水平同期信号 (Hsync) 及び垂直同期信号 (Vsync) とが供給される。これら水平同期信号 (Hsync) 及び垂直同期信号 (Vsync) が供給されない場合には、内部でこれらを表示用画素クロック (pelCLK) から生成する。そして、これらの信号に基づいてタイミング・制御信号とを生成して各処理ブロック 5 ~ 11 に送り、これら処理ブロックが互いに協調して復号化処理を行なうように制御する。

【0047】図 2 は図 1 における外部メモリ 2 の内部構成の一具体例を示す模式図である。

【0048】同図において、外部メモリ 2 は 3 つのフレームメモリ領域 14 ~ 16 と符号化データ用バッファ領域 17 を有し、これらフレームメモリ領域 14, 15, 16 には夫々復号化画像を 1 つずつ記憶するようにし、符号化データ用バッファ領域 17 には、入力バッファメモリ 5 (図 1) からの符号化データを一時記憶するようとする。これら領域 14 ~ 17 での書き込み / 読出制御がメモリコントローラ 11 によって行なわれる。

【0049】図 3 はかかるメモリコントローラ 11 の一具体例を示すブロック図であって、11a は符号化データ用バッファ制御部、11b は参照画像読出制御部、11c は復号化画像書込制御部、11d は復号化画像読出

制御部、11e はバス幅変換部である。

【0050】同図において、符号化データ用バッファ制御部 11a は、外部メモリ 2 内の符号化データ用バッファ領域 17 (図 2) に空きがあると、入力バッファメモリ 5 (図 1) にデータリクエスト信号を送り、それに応じて送られてきた符号化データをバス変換部 11e で処理した後、この符号化データ用バッファ領域 17 に格納するように、外部メモリ 2 へアドレス及び制御信号を出力する。また、この符号化データ用バッファ制御部 11a は、復号用バッファメモリ 6 (図 1) からデータリクエスト信号を受けると、それに応じて符号化データ用バッファ領域 17 (図 2) から符号化データを読み出し、バス変換部 11e で処理した後、復号用バッファメモリ 6 に転送する。

【0051】参照画像読出制御部 11b は、動き補償ユニット 9 (図 1) から動きベクトル信号を受け、それに基づいて外部メモリ 2 にアドレス及び制御信号を出力し、そのフレームメモリ領域 14 ~ 16 (図 2) から所定の参照画像を読み出す。

【0052】復号化画像書込制御部 11c は、動き補償ユニット 9 (図 1) からの復号化画像を外部メモリ 2 でのフレームメモリ領域 14 ~ 16 (図 2) のいずれかに格納するように、外部メモリ 2 へアドレス及び制御信号を出力する。また、復号化画像読出制御部 11d は、表示のタイミングに合わせて、これらフレームメモリ領域 14 ~ 16 のいずれからか所望とする復号化画像を読み出すためのアドレス及び制御信号を出力する。

【0053】図 4 は復号化処理タイミングと外部メモリ 2 内のフレームメモリ領域 14 ~ 16 での復号化画像の書き込み / 読出し及び表示タイミングとの関係を示すタイミングチャートであって、Vsync は表示系の垂直同期信号を示す。また、ここでは、図 13 で示した符号化映像信号を復号するものとし、符号化データはピクチャが I 1, P 5, B 2, B 3, B 4, P 9, B 6, B 7, B 8 の順となっている。ここで、勿論、I 1 は I ピクチャ、P 5, P 9 は P ピクチャ、B 2 ~ B 4, B 6 ~ B 8 は B ピクチャであり、それらの数字は表示の順序を示している。

【0054】図 2 において、外部メモリ 2 では、フレームメモリ領域 14, 15 が参照画像としての I ピクチャまたは P ピクチャの書き込みに割り当てられ、フレームメモリ領域 16 が B ピクチャの書き込みに割り当てられる。ここでは、各ピクチャが 1 フレーム分の画像から構成されているものとする。

【0055】最初の 1 フレーム期間で I 1 ピクチャの復号化処理が行なわれ、その復号化画像 I 1 (以下、復号前のピクチャと同じ符号を用いる) はフレームメモリ領域 14 に書き込まれる。次の 1 フレーム期間では、復号化画像 I 1 をフレームメモリ領域 14 から読み出し、これを参照画像として P 5 ピクチャの復号化処理を行な

い、この結果得られた復号化画像 P 5 をフレームメモリ領域 1 5 に書き込む。そして、さらに次の 1 フレーム期間では、これら復号化画像 I 1, P 5 を読み出して参照画像とし、B 2 ピクチャの復号化処理を行なう。これによって得られる復号化画像 B 2 はフレームメモリ領域 1 6 に書き込まれる。以下同様にして、各ピクチャの復号化処理が行なわれる。

【0056】次に、表示のための外部メモリ 2 からの読み出しついて説明する。

【0057】あるピクチャの復号化画像のフレームメモリ領域 1 4, 1 5 または 1 6 への書き込みが行なわれるごとに、次に、この同じフレームメモリ領域に新たな復号化画像が書き込まれる前にこの読み出しが行なわれなければならない。例えば、図 4 で B 2 ピクチャに着目すると、この B 2 ピクチャが復号化処理されてフレームメモリ領域 1 6 に書き込まれた後、引き続き次の 1 フレーム期間では、B 3 ピクチャが復号化処理されてフレームメモリ領域 1 6 に書き込まれることになる。

【0058】ここで、もし、復号化画像 B 2 の表示のための読み出しが開始を B 2 ピクチャの復号化処理が完全に終了するまで待たせたとすると、この読み出しが次の B 3 ピクチャの復号化及びその復号化画像 B 3 のフレームメモリ領域 1 6 への書き込みと重なってしまうので、フレームメモリ領域がもう 1 つ余分に用意し、復号化画像 B 2, B 3 を夫々別々のフレームメモリ領域に格納するようしなければならない。

【0059】しかし、実際には、B 2 ピクチャの復号化処理（書き込み）が完全に終了するまでその表示（読み出し）を待つ必要はなく、復号化処理が済んだ部分から後追いで順次読み出していけばよい。このようにすることにより、B ピクチャ用のフレームメモリ領域として 1 フレーム分あるだけで、B ピクチャの復号化処理と表示が問題なく行なえることになる。

【0060】但し、符号化映像信号では、1 ピクチャ内に符号量の偏りがあるので、復号化処理の進み具合は局的にムラが存在する。そこで、この実施例においては、図 4 に示したように、B 2 ピクチャの復号化処理（書き込み）開始から 1 フィールド遅れて復号化画像 B 2 の表示（読み出し）を開始するようにして、マージンを取っている。

【0061】以上のように、外部メモリ 2 の容量を最小限にして効率的に利用しながら復号化処理を行なうためには、表示系の垂直同期信号 Vsync に同期して復号化処理を制御することが不可欠であり、表示系の垂直同期信号 Vsync のタイミングを無視した復号化処理は現実的でない。

【0062】図 1 におけるタイミングユニット 1 2 は、以上のように、復号化処理を制御するものであって、このためのタイミング・制御信号を生成して出力する。かかるタイミング・制御信号を図 5 により説明する。但

し、ここでは、復号化画像による映像信号が N T S C 方式のテレビジョン信号に準拠した映像信号であるものとし、図中の Vsync, Hsync は夫々復号化画像を表示するための垂直、水平同期信号である。

【0063】かかるタイミング・制御信号は、図 1 5 で示したピクチャレイヤ以上の階層の復号化処理を開始することを示す制御信号(PictStart) と、スライスレイヤを含む 1 マクロブロックの復号化を開始することを示す制御信号(MbStart) とからなっている。

【0064】復号化しようとするピクチャが 1 フレームに相当する場合、図 5 (1) に示すように、1 フレーム毎に制御信号(PictStart) を 1 個、制御信号(MbStart) を 1 フレーム分のマクロブロック数を復号化するのに充分な個数だけ夫々出力する。同様に、復号化しようとするピクチャが 1 フィールドに相当する場合には、図 5 (2) に示すように、1 フィールド毎に制御信号(PictStart) を 1 個、制御信号(MbStart) を 1 フィールド分のマクロブロック数を復号化するのに充分な個数だけ出力する。図 5 (1), (2) は、復号化しようとするピクチャがフレーム画像であるか、フィールド画像であるかに応じて適応的に変える。

【0065】図 6 は図 1 における可変長復号ユニット 7 の一具体例を示すブロック図であって、7 a はバレルシフタ、7 b は復号化テーブル、7 c はバッファメモリ、7 d はバッファメモリ制御器、7 e はスタートコード検出器、7 f はタイミング制御器、7 g は復号制御器である。

【0066】同図において、復号用バッファメモリ 6 (図 1) からの符号化データは、バレルシフタ 7 a に供給される。このバレルシフタ 7 a は、取り込んだ符号化データの一部を復号化テーブル 7 b に供給する。復号化しようとする符号は、一般に、可変長符号であるので、この復号化テーブル 7 b からその符号長を受けとて次の符号データの頭出しを行なう。バレルシフタ 7 a 内の符号化データを使い切ると、復号用バッファメモリ 6 (図 1) に対してデータリクエスト信号を送り、次の符号化データを要求する。

【0067】復号化テーブル 7 b は、符号化データに含まれるパラメータと D C T 係数とを復号化するものであって、図 7 に示すように、n 個のテーブルからなり、テーブル 1 は D C T 係数を、テーブル 2 ~ n はパラメータや符号長などの D C T 係数以外の特定の符号を夫々復号化するためのものである。後述する復号制御信号により、それらのうちのどれを選択するか決定される。

【0068】なお、供給された符号化データに該当する符号が選択されたテーブル中に存在しない場合には、エラーが発生したと判断してエラー検出信号を出力する。

【0069】図 6 において、復号化テーブル 7 b から得られる夫々の符号の符号長データはバレルシフタ 7 a に供給される。また、復号化テーブル 7 b から出力された

データのうちのD C T 係数は一旦バッファメモリ 7 c に格納され、1 ブロック分である 6 4 個のD C T 係数が揃うと、I D C T ユニット 8 (図 1) に出力する。

【0 0 7 0】バッファメモリ制御器 7 d は内部にカウンタを備え、復号化テーブル 7 b から出力されるD C T 係数の個数をカウントしてバッファメモリ制御信号を出し、上記のように、バッファメモリ 7 c を制御する。万一D C T 係数が 6 4 個より多く送られた場合には、エラーが発生したと判断してエラー検出信号を出力する。

【0 0 7 1】スタートコード検出器 7 e は、符号化データが供給され、その中からスタートコード (図 1 5 を参照) に特有なパターンを検出すると、スタートコード検出信号を出力する。

【0 0 7 2】以上のような可変長復号ユニット 7 の動作を制御するのが、タイミング制御器 7 f と復号制御器 7 g である。

【0 0 7 3】図 8 はこのタイミング制御器 7 f の一具体例を示すブロック図であって、7 f 1 は復号タイミング制御部、7 f 2 はMBアドレスカウント部、7 f 3 はMBアドレス抽出部、7 f 4 はMBアドレス比較制御部である。

【0 0 7 4】同図において、復号タイミング制御部 7 f 1 は、タイミングユニット 1 2 (図 1) からタイミング・制御信号が供給され、復号制御器 7 g (図 6) を始めとする可変長復号ユニット 7 内の処理ブロックに適切なタイミングで動作開始及び停止の指示を与える復号タイミング制御信号を生成出力する。即ち、タイミング・制御信号の制御信号(PictStart) によってピクチャレイヤの開始が指示され、また、制御信号(MbStart) によってマクロブロックの開始が指示されて、この復号タイミング制御信号が生成される。

【0 0 7 5】MBアドレスカウント部 7 f 2 は、上記タイミング・制御信号のうちの制御信号(PictStart) によってクリアされ、制御信号(MbStart) をカウントすることにより、復号化タイミング上のマクロブロックアドレスを生成する。一方、MBアドレス抽出部 7 f 3 は、復号化テーブル 7 b (図 6) で符号化データを解析した結果得られるパラメータを基に、符号化データ上のマクロブロックアドレスを抽出して再生する。復号化処理はタイミング・制御信号をタイミング基準として行なわれ、正常の復号化処理では、タイミング・制御信号の制御信号(MbStart) が制御信号 (P i c t S t a r t) から何個目であるのものであるかで決まるマクロブロックアドレスのマクロブロックが処理される。このため、復号化処理が正常に行なわれている場合には、これら 2 つのマクロブロックアドレスは一致するはずである。そこで、MBアドレス比較制御部 7 f 4 で両者が比較され、もし後者 (即ち、符号化データから抽出されたマクロブロックアドレス) が遅れている場合には、エラーが発生したと判断してエラー検出信号を出力する。もし後者が進み

すぎていた場合には、一致するまでの間このマクロブロックの復号化処理を一時停止させる復号タイミング制御信号を出力する。

【0 0 7 6】図 9 は図 6 における復号制御器 7 g の一具体例を示すブロック図であって、7 g 1 は復号状態カウンタ、7 g 2 は分岐先計算部、7 g 3 はエラー検出器、7 g 4 はエラーレベルカウンタである。

【0 0 7 7】符号化データでは、符号化方式の規格で予め定義された符号が決められた順番で配列されている。

10 但し、この順番は、ただ単純に符号が並んでいるだけではなく、条件に応じて特定の符号の配列順や存在が変化する。そして、その条件自体を表わすデータも符号化データ中に符号化されている。

【0 0 7 8】このような性質を有する符号化データを適切に復号化処理するために、この実施例では、図 9 に示すように、復号制御器 7 g 内に復号状態カウンタ 7 g 1 を設け、それによって上記符号の順番を含めた復号化処理の状態を管理、制御するようにしている。復号状態カウンタ 7 g 1 で保持される値は現在の復号状態を示しており、それは、同時に、復号制御信号として可変長復号ユニット 7 内の各処理ブロック (図 6) の制御に用いられる。

【0 0 7 9】分岐先計算部 7 g 2 は、上記復号制御信号が示す現在の復号状態とその時点までに復号化されたパラメータの値に基づいて次の復号状態を決定し、これを復号状態カウンタ 7 g 1 に指示する。このようにして、符号化データ中に含まれている条件データに基づいて、適切な順番で符号を解析することが可能となる。

【0 0 8 0】なお、復号化テーブル 7 b (図 7)、バッファメモリ 7 d、タイミング制御器 7 f (図 6、図 8)、または、後述のエラー検出器 7 g 3 からエラー検出信号を受けると、エラー処理を行なうルーチンに分岐するように復号状態カウンタ 7 g 1 に指示する。さらに、符号化データの規格上決められた位置以外でスタートコードを検出した場合も、エラーと判断してエラー処理ルーチンに分岐させる。

【0 0 8 1】エラー検出器 7 g 3 は、復号化テーブル 7 b (図 6) で復号化されたパラメータの値を調べ、それが予め決められた範囲内に収まっているかどうかをチェックする。もし収まっていない場合には、エラーが発生したと判断してエラー検出信号を出力する。また、符号化データ中にシーケンスエラーコードを検出した場合でも、やはりエラーが発生したと判断してエラー検出信号を出力する。

【0 0 8 2】エラーレベルカウンタ 7 g 4 は、エラー検出によって復号制御器 7 g がエラー処理ルーチンに分岐した場合、復号状態カウンタ 7 g 1 からの復号制御信号によってそのエラーレベルを判断する。この判断結果を表わすエラーレベル信号は復号化装置 1 内の各処理ブロックに分配される。

【0083】次に、この復号制御器7gでのエラー発生時の処理についてさらに詳しく説明する。

【0084】上記のように、エラー検出方法のいずれかによってエラーが検出されると、分岐先計算部7g2の指示でエラー処理ルーチンに分岐する。このエラー処理ルーチンの一具体例を図10によって説明する。

【0085】まず、エラーを検出する（ステップ101）と、エラーレベルを1に設定する（ステップ102）。具体的には、エラーレベルカウンタ7g4（図9）が復号状態を復号状態カウンタ7g1（図6）からの復号制御信号から検知して、エラーレベルを1に設定する（以下、エラーレベルの設定は同様の仕組みで行なわれる）。

【0086】次に、この時点で上記制御信号（PictStart）（これは、復号の垂直同期パルスとみられるものである）を検出したかどうかを調べ（ステップ103）、検出していなければ、正常な復号化処理へ復帰するきっかけとなるスタートコードを検出したかどうかを調べる（ステップ104）。スタートコードを検出した場合、即ち、復号の垂直同期信号（PictStart）が検出する前にスタートコードを検出した場合、このエラーは格別大きな影響力がなく、重大性が低いものとして、正常に復号化処理への復帰処理を行なう（ステップ103）。

【0087】復号の垂直同期信号（PictStart）もスタートコードも検出されないときには、ステップ103に戻り、復号の垂直同期信号（PictStart）、スタートコードのいずれかを検出するまで、ステップ103、104の動作が繰り返される。

【0088】上記復帰処理のルーチン（ステップ113）は、スライスレイヤ以上の階層で復帰を図るルーチンである。このように、エラーレベルが1の場合には、符号化データの階層構造のうちのスタートコードを有するレイヤとしては最下層であるスライスで復帰を図ることができる。

【0089】スタートコードを検出するより前に復号の垂直同期信号（PictStart）を検出した場合には、エラー中に他のピクチャが移ってしまったなどの事態が生じ、エラーレベル1のエラーよりも重大なものとなる。このため、かかるエラーに対しては、エラーレベルを2に設定し（ステップ106）、先のステップ103、104と同様に、復号の垂直同期信号（PictStart）、スタートコードのいずれかを検出する処理を繰り返す（ステップ107、108）。

【0090】この場合、復号の垂直同期信号（PictStart）よりも先にスタートコードを検出した場合には、さらに、それがピクチャヘッダ以上であるかどうかを調べ（ステップ109）、そうであるならば、復帰処理を行ない（ステップ114）、ピクチャヘッダより下位のレ

イヤ、即ち、スライスヘッダである場合には、再度ステップ107に戻るようしている。

【0091】このようにして、エラーレベルが2の場合には、ピクチャレイヤ以上の階層の開始から復帰を図るようにすることが可能である。

【0092】さらに、スタートコードよりも先に復号の垂直同期信号（PictStart）を検出した場合には（ステップ107）、ピクチャの開始さえ検出されないような重大性が最も高いエラーが生じたことになり、このようなエラーに対しては、エラーレベルを3に設定する（ステップ110）。そして、スタートコードを検出するまで待ち（ステップ111）、スタートコードを検出すると、それがシーケンスヘッダ以上であるかどうかを調べ（ステップ112）、そうであるならば、復帰処理を行ない（ステップ115）、シーケンスレイヤより下位のレイヤのヘッダである場合には、再度ステップ111に戻るようしている。

【0093】このようにして、エラーレベルが3の場合には、シーケンスレイヤ以上ののみで復帰を図るようにすることができる。

【0094】以上のようにして得られるエラーレベル信号は、復号化装置1内の図1に示す各ブロックに分配されている。

【0095】次に、エラー発生時のエラー隠し処理について説明する。

【0096】図1において、エラーレベルが1の場合には、動き補償ユニット9で、通常の処理によってマクロブロックの画像を復号する代わりに、現在処理中のマクロブロックと同じ位置の直前の参照画像のデータを用いてこのマクロブロックのエラーを隠す。そのためには、メモリコントローラ11にゼロの動きベクトルを与えて参照画像を読み出させる。このようにして、最も軽微なエラーであるレベル1の場合、エラー隠しはマクロブロック単位で行なわれる所以、その後正常な復号化処理に復帰した場合でも、エラーが画面上に与える影響は最小限で済むことになる。

【0097】エラーレベルが2の場合には、表示ユニット10で、現在復号中のピクチャを表示する代わりに、既に復号済みの直前のピクチャ（参照画像）を表示するようにしてエラーを隠す。そのためには、メモリコントローラ11に送る表示画像を指定する信号を、エラー時には、直前の参照画像を指定するように切り替える。このようにして、エラーレベル2の場合には、ピクチャ単位でエラー隠しを行なうようにすることができる。

【0098】また、エラーレベルが3の場合には、最も重大なエラーであって、それ以上そのまま復号を継続することが困難であると判断し、一旦入力バッファメモリ5や復号用バッファメモリ6、メモリ2の符号化データ用バッファ領域17（図2）をクリアする。この符号化データ用バッファ領域17のクリアは、メモリコントロ

ーラ 11 内の符号化データ用バッファ制御部 11 a (図 3) にエラーレベルが 3 のエラーレベル信号を送ることによって行なわれる。その後、新たに入力端子 3 から入力された符号化データに対して復号化を試みる。また、可変長復号ユニット 7 がシーケンスヘッダを発見して正常な復号化処理に復帰するまでの間、表示ユニット 10 は既に復号済みの直前のピクチャ (参照画像) を表示し続けるようにする。

【0099】このようにして、この実施例では、エラーから復帰するまでの時間でエラーレベルを複数の段階に評価をするものであって、これにより、夫々のエラーレベルに応じて復帰させる過程を異ならせることが可能である。また、夫々のエラーレベルに応じた適切なエラー隠しを行なうことができるので、エラーの影響で画像が乱れる時間を必要最低限に抑えることが可能である。

【0100】図 11 は図 6 での復号制御器 7 g の他の具体例を示すブロック図であって、7 g 5 はエラー回数カウンタであり、図 9 に対応する部分には同一符号を付けて重複する説明を省略する。

【0101】同図において、この具体例は、図 9 に示した構成にエラー回数カウンタ 7 g 5 が追加されたものであり、このエラー回数カウンタ 7 g 5 は、タイミング制御器 7 f (図 6) からエラー検出信号を受けると、そのカウンタ値を 1 だけ増加させてエラー回数を計数し、その計数値を分岐先計算部 7 g 2 に与える。

【0102】次に、図 12 により、図 11 に示す復号制御器 7 g でのエラー処理ルーチンの一具体例について説明する。

【0103】このエラー処理ルーチンが図 10 に示したものと異なる点は、エラーレベルの評価を復号の復号の垂直同期信号(PictStart) という時間に基づいた信号で行なうのではなく、過去にエラーの発生した回数で行なうようにした点にある。

【0104】まず、エラー検出方法のいずれかによりエラーが検出されると、エラー回数カウンタ 7 g 5 のカウント値が 1 だけ増加するとともに、エラー処理ルーチンに分岐する (ステップ 201)。ここで、過去にエラーが発生した回数を調べ、それが 1 回 (即ち、今回が初めて) であるならば、エラーレベルを 1 に設定し (ステップ 203)、スタートコードを検出するまで待機する (ステップ 204)。そして、スタートコードを検出すると、復帰処理ルーチン (ステップ 213) に映る。

【0105】この復帰処理ルーチン (ステップ 213) は、スライスレイヤ以上で復帰を図るルーチンであり、エラーレベルが 1 の場合には、符号化データの階層構造のうちのスタートコードを有するレイヤとしては最下層であるスライスで復帰を図ることができる。

【0106】一方、エラー発生回数が 1 回より多く (ステップ 202)、かつ、過去のエラー発生回数が 2 回 (即ち、今回が 2 回目) であるときには (ステップ 20

6)、エラーレベルを 2 に設定し (ステップ 207)、スタートコードを検出するまで待機する (ステップ 208)。そして、スタートコードを検出すると、それがピクチャヘッダ以上であるかどうかを調べ (ステップ 209)、そうであれば、復帰処理に移行し (ステップ 214)、ピクチャヘッダより下位のレイヤ、即ち、スライスヘッダである場合には、再度ステップ 208 に戻るようにしている。この点がエラーレベルが 1 のときと異なる。

【0107】このようにして、エラーレベルが 2 の場合には、ピクチャレイヤ以上のみで復帰を図るようにすることが可能である。

【0108】また、エラー発生回数が 2 回より多い場合には (ステップ 202)、エラーレベルを 3 に設定し (ステップ 210)、スタートコードを検出するまで待機する (ステップ 211)。スタートコードを検出した場合には、さらに、それがシーケンスヘッダ以上であるかどうかを調べ (ステップ 212)、そうであれば、復帰処理に移行し (ステップ 215)、シーケンスレイヤより下位のレイヤのヘッダであった場合には、再度ステップ 211 に戻るようにしている。

【0109】このようにして、エラーレベルが 3 の場合には、シーケンスレイヤ以上のみで復帰を図るようにすることが可能である。

【0110】なお、エラーレベルに応じたエラー隠し処理については、先に説明したものと同様である。

【0111】以上のように、この具体例では、エラーが発生した回数でエラーレベルを複数の段階に評価するものであり、夫々のエラーレベルに応じて復帰させる過程を異ならせることが可能である。また、夫々のエラーレベルに応じた適切なエラー隠しを行なうことができるので、エラーの影響で画像が乱れる時間を必要最低限に抑えることが可能である。

【0112】以上、本発明の実施例について説明したが、本発明はかかる実施例のみに限定されるものではない。

【0113】即ち、上記実施例では、入力符号化データを MPEG に準じたものとしたが、これに限らず、同様の性質を備えた別の符号化方法で符号化されたデータであってもよい。

【0114】また、図 4 において、上記実施例における復号化処理とメモリ 2 内に設けられたフレームメモリ領域 14 ~ 16 の利用及び表示のタイミングの関係を、各ピクチャが 1 フレーム分の画像から構成されている場合を例として説明したが、これに限定されるものではなく、各ピクチャが 1 フィールド分の画像から構成されている場合についても同様である。

【0115】さらに、図 5 において、タイミングユニット 12 (図 1) からの制御信号を NTSC 方式のテレビジョン信号を例として説明したが、これに限定されるも

のではなく、他の方式のテレビジョン信号でも同様である。

【0116】さらにまた、上記実施例においては、エラーレベルを3段階に評価するものとしたが、これに限定されるものではなく、さらに細かくあるいは逆に2段階にしてもよく、必要に応じて変えてよい。

【0117】さらにまた、図10に示したエラー処理ルーチンでは、エラーレベルを評価するための信号として復号の垂直同期信号(PicStart)を用いたが、これに限定されるものではなく、表示系の垂直同期パルス(Vsync)そのものあるいは他のこれに密接に関係する信号を用いるようにしてもよい。

【0118】さらにまた、上記実施例においては、エラーレベルが2以上の場合、既に復号済みのピクチャ(参照画像)を表示するようにしたが、これに限定されるものではなく、例えば無地の画像や他の任意の画像を表示するようにしてもよい。

【0119】

【発明の効果】以上説明したように、本発明によれば、エラーから復帰するまでの時間によりあるいはエラーが発生した回数により、エラーレベルを複数の段階に評価し、夫々のエラーレベルに応じて復帰させる過程を異ならせることが可能であって、夫々のエラーレベルに応じた適切なエラー隠しを行なうことができるので、エラーの影響で画像が乱れる時間を必要最低限に抑えることができる。

【図面の簡単な説明】

【図1】本発明による符号化映像信号の復号化処理方法及びそれを用いた復号化装置の一実施例を示すブロック図である。

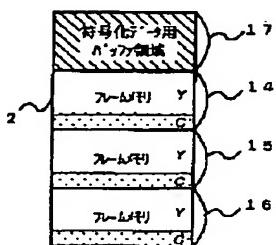
【図2】図1におけるメモリの内部構成の一具体例を示す模式図である。

【図3】図1におけるメモリコントローラの一具体例を示すブロック図である。

【図4】図1に示した実施例での復号化処理タイミングとメモリ内のフレームメモリ領域での復号画像の書き込み／読み出し及び表示タイミングとの関係を示すタイミング

【図2】

【図2】



チャートである。

【図5】図1におけるタイミングユニットから出力される制御信号の一具体例を示す図である。

【図6】図1における可変長復号ユニットの一具体例を示すブロック図である。

【図7】図6における復号化テーブルの一具体例を示すブロック図である。

【図8】図6におけるタイミング制御器の一具体例を示すブロック図である。

10 【図9】図6における復号制御器の一具体例を示すブロック図である。

【図10】図9に示した復号制御器で指示されるエラー処理ルーチンの一具体例を示すフローチャートである。

【図11】図6における復号制御器の他の具体例を示すブロック図である。

【図12】図11に示した復号制御器で指示されるエラー処理ルーチンの一具体例を示すフローチャートである。

20 【図13】MPEGに準ずる符号化コードの一例を示す図である。

【図14】図13で示した符号化データでのマクロブロックの構成を示す図である。

【図15】MPEGの符号データでのレイヤの階層構造を示す図である。

【符号の説明】

1 復号化装置

2 メモリ

3 入力端子

4 出力端子

30 5 入力バッファメモリ

6 復号用バッファメモリ

7 可変長復号ユニット

8 IDCTユニット

9 動き補償ユニット

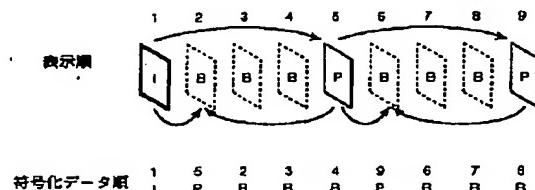
10 表示ユニット

11 メモリコントローラ

12 タイミングユニット

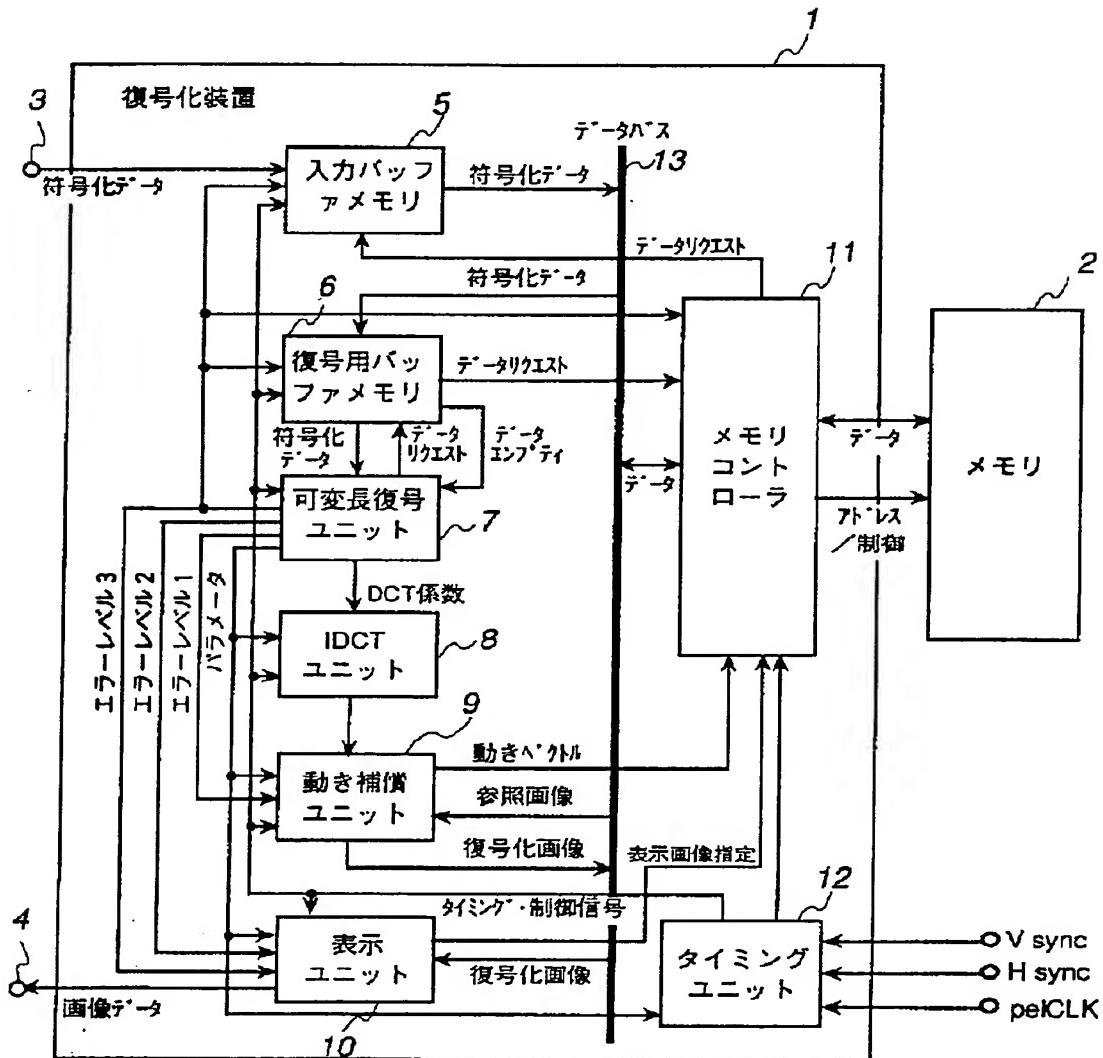
【図13】

【図13】



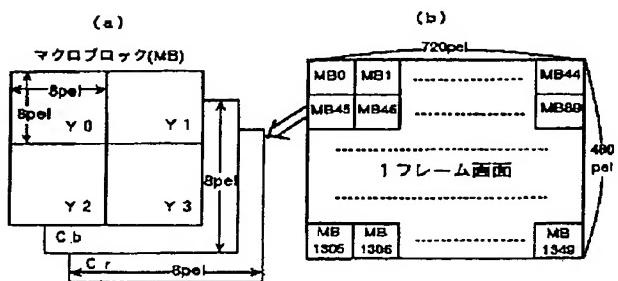
【図 1】

【図 1】



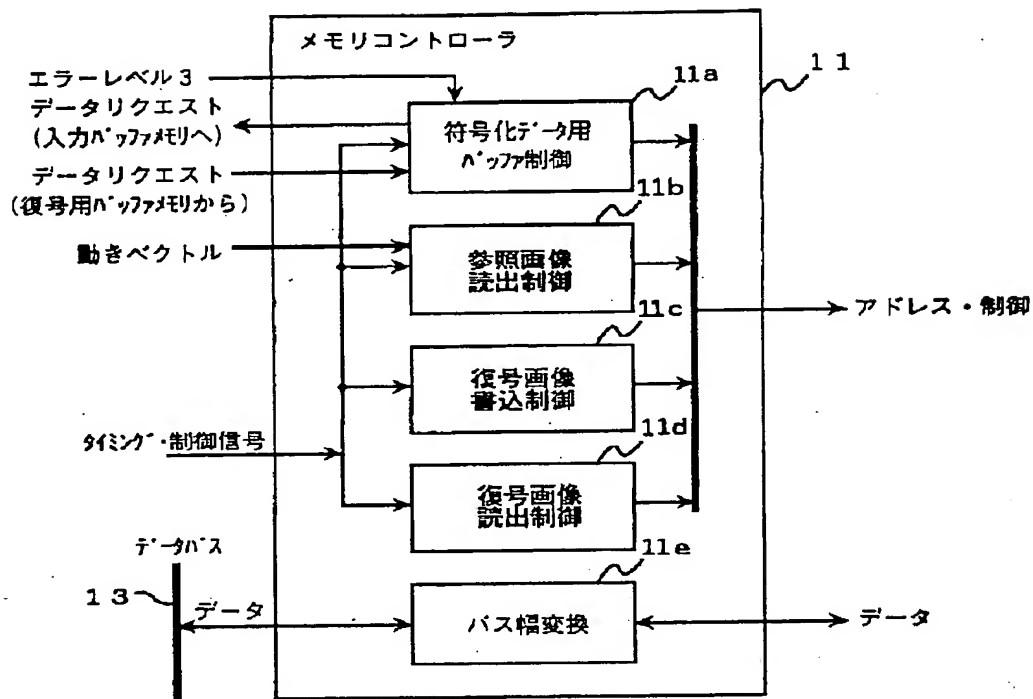
【図 1-4】

【図 1-4】



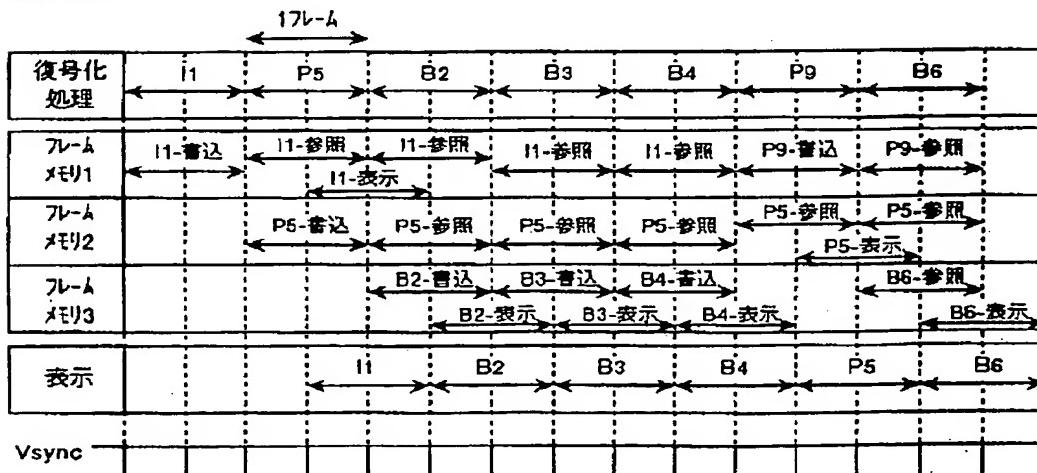
【図 3】

【図 3】



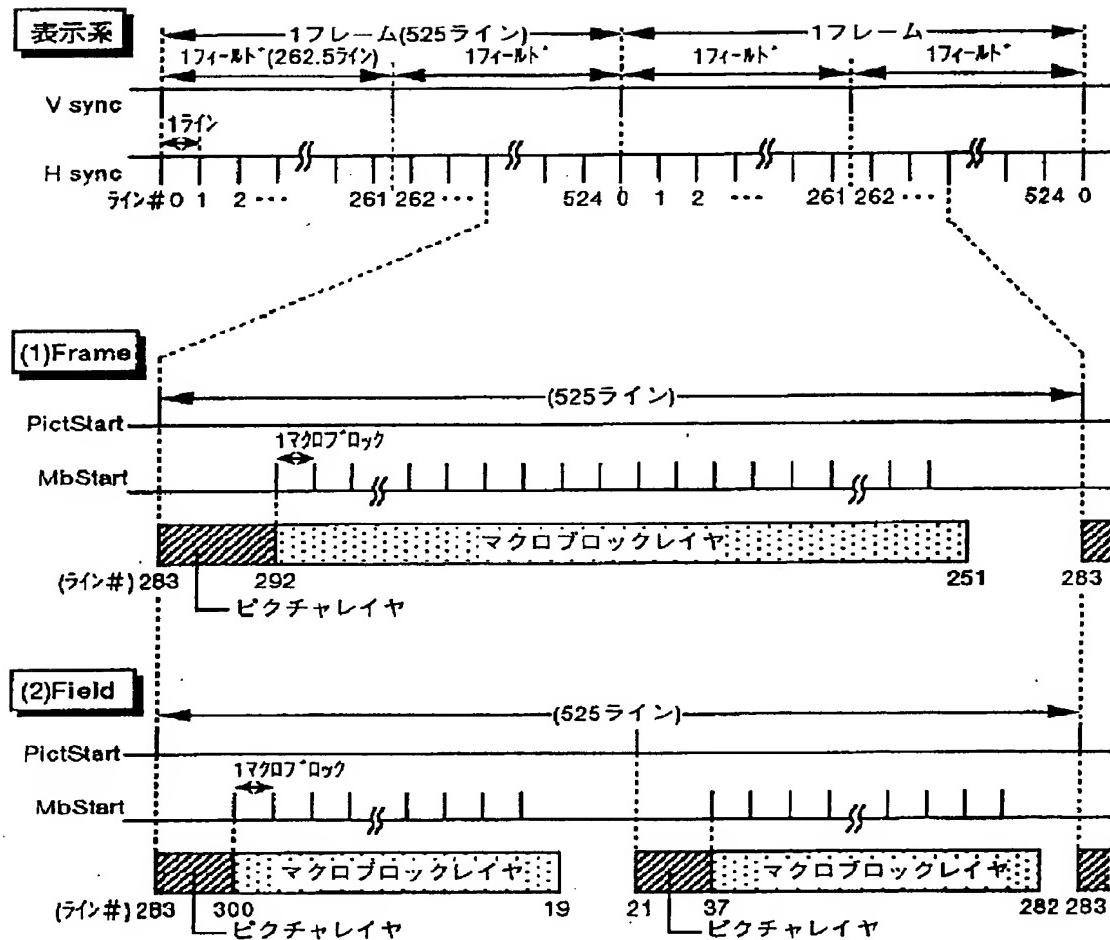
【図 4】

【図 4】



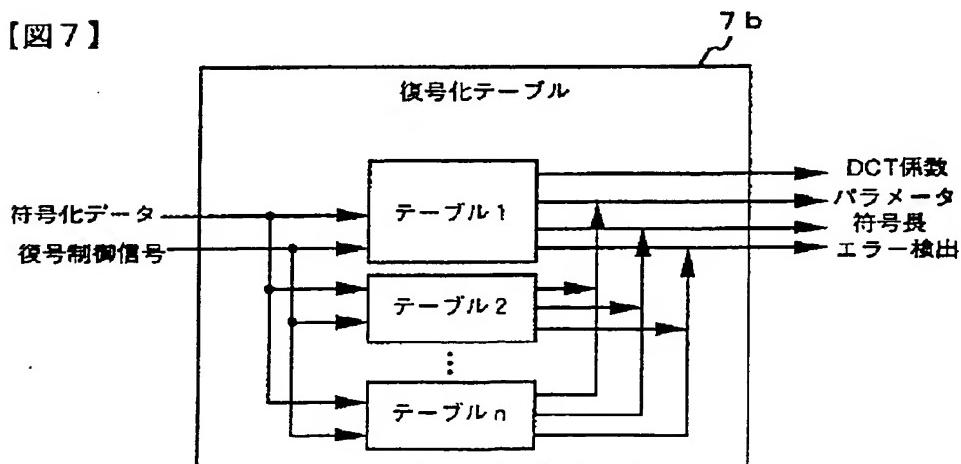
【図 5】

【図 5】



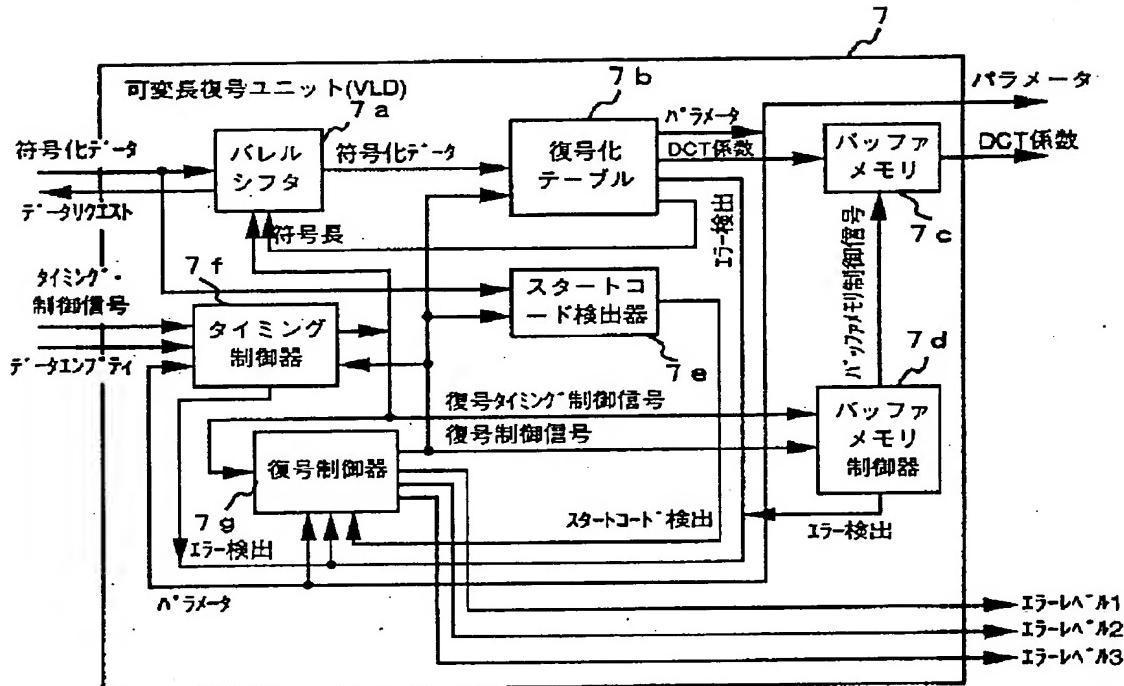
【図 7】

【図 7】



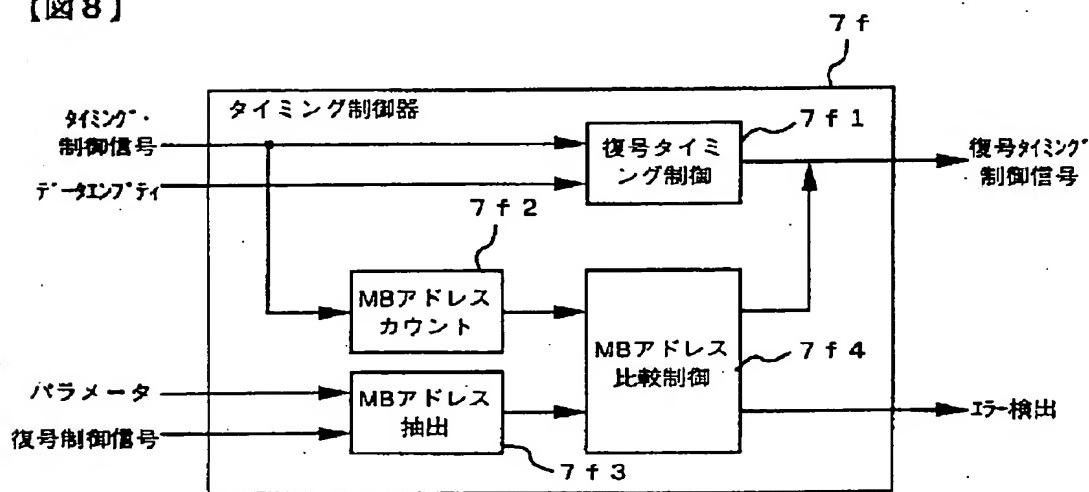
【図 6】

【図 6】



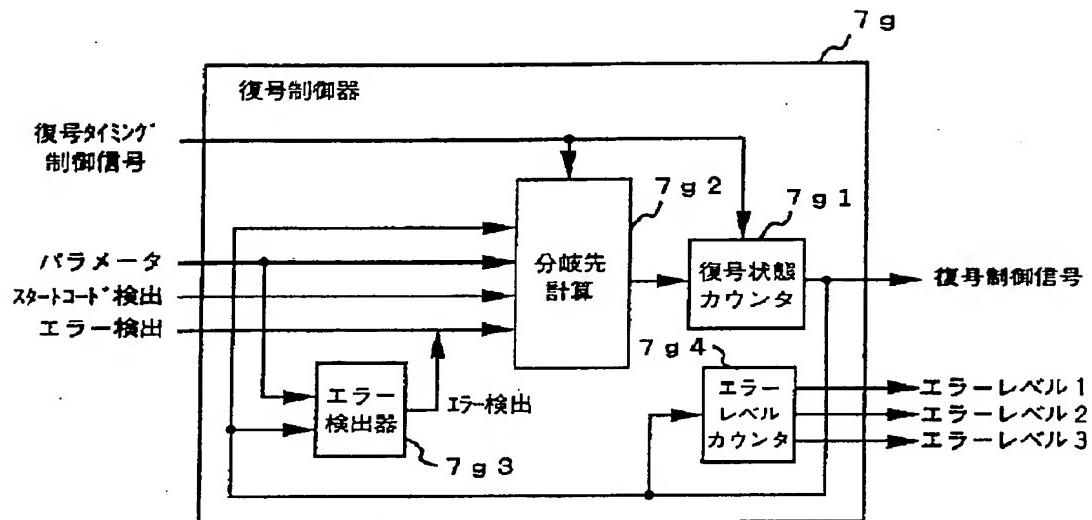
【図 8】

【図 8】



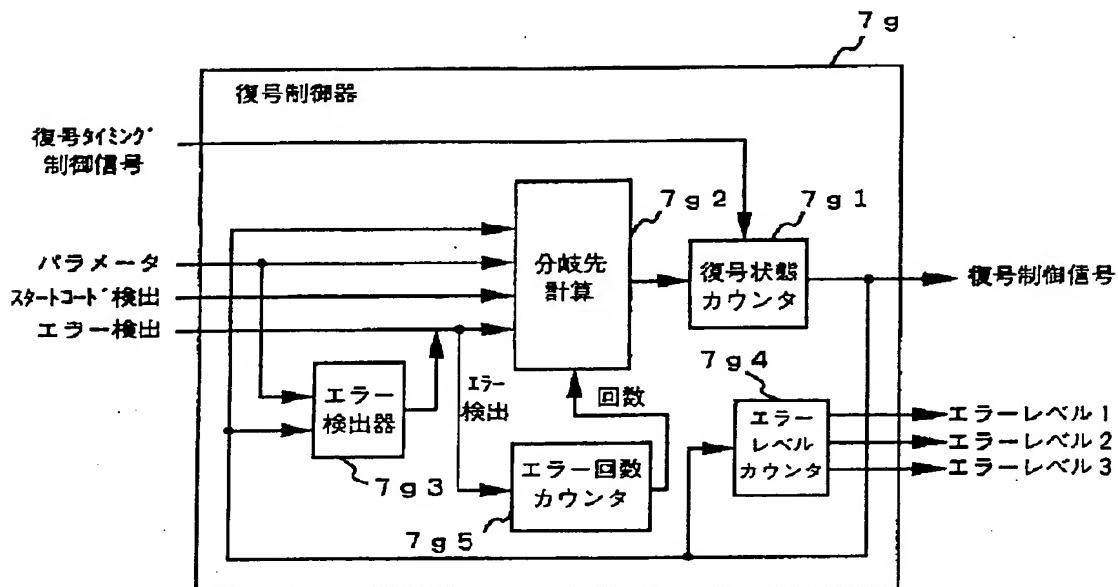
【図 9】

【図 9】



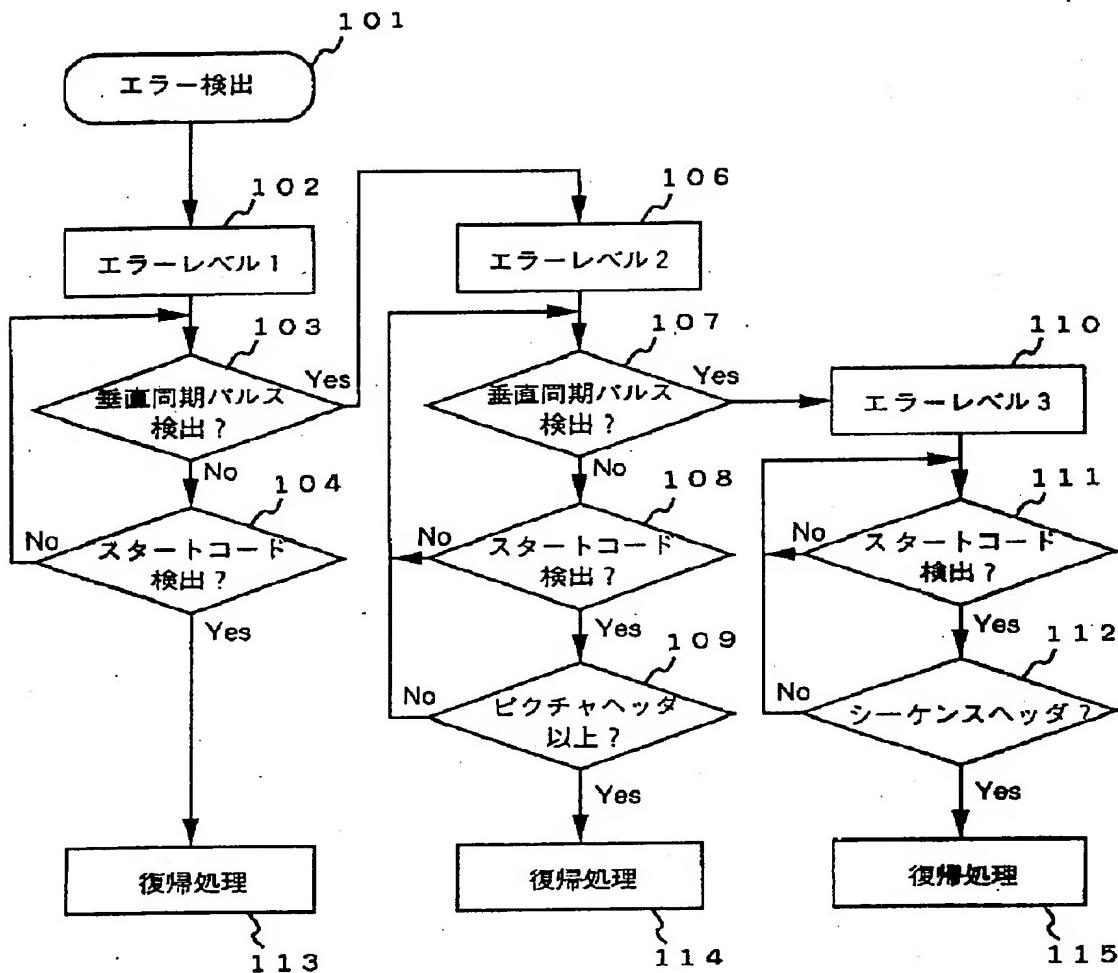
【図 11】

【図 11】



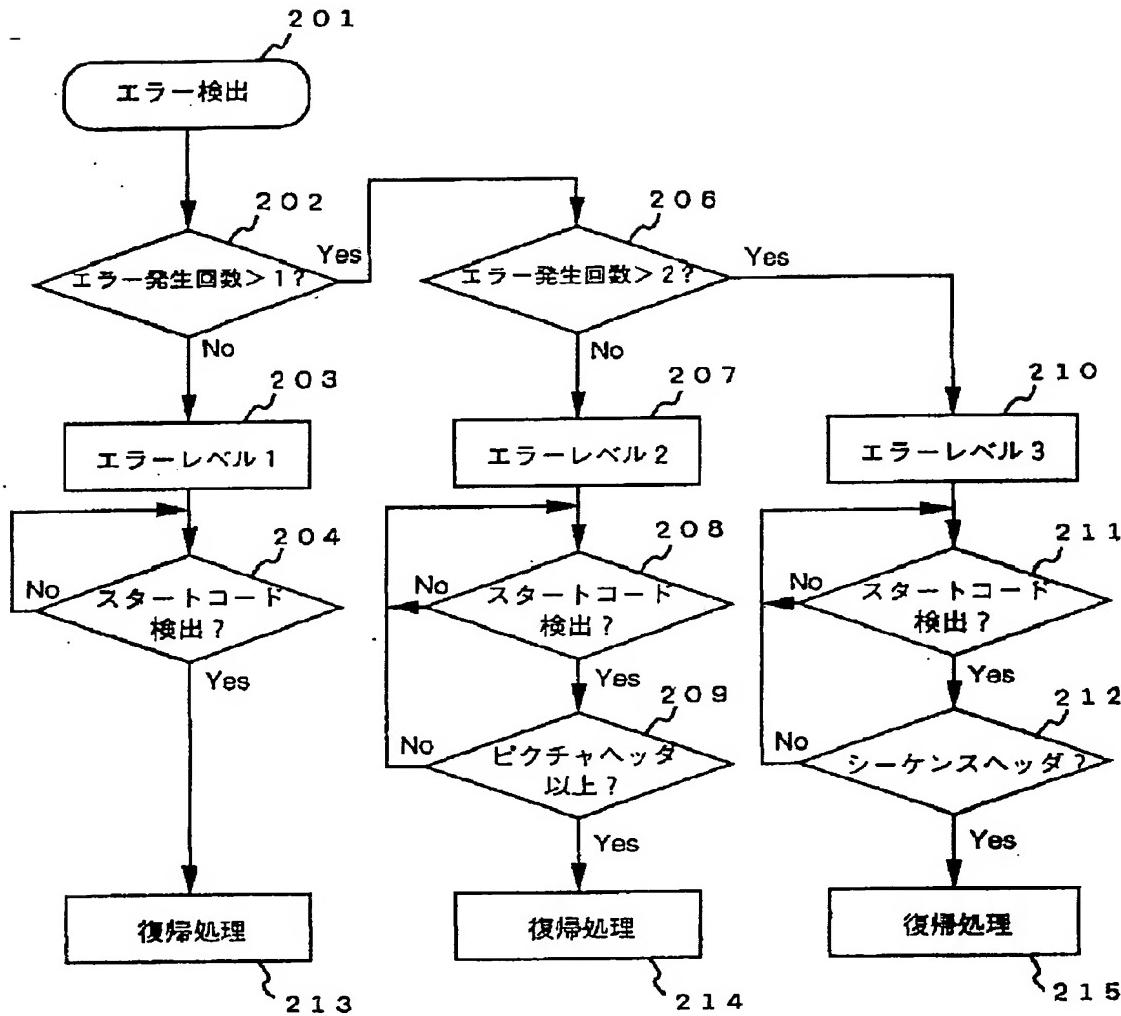
【図 10】

【図 10】



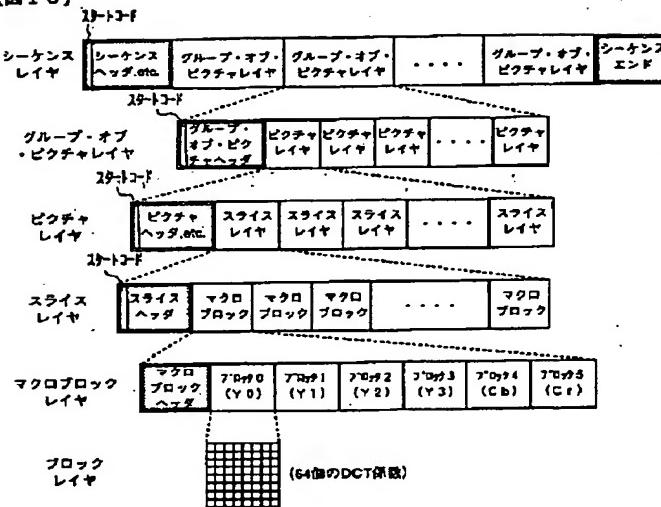
【図 12】

【図 12】



【図 15】

[図 15]



THIS PAGE BLANK (USPTO)